5. Dylan

Program Name: Dylan.java Input File: dylan.dat

Dylan is traveling on a bus to the grand Hilbert Hotel! Everything's bigger in Texas, and this bus is no exception. The bus has a seemingly infinite number of rows, and there are 2 * K seats per row. The seats in a row are numbered 1 through 2 * K, and a center aisle separates seats K and K + 1. The rows are numbered row 1, then row 2, and so on.

When disembarking the bus, people get out one row at a time. The person closest to the center aisle on the left side gets out first, then the person on the right side, and so on. For example, when K = 3, the seats in a row are 1 2 3 4 5 6, with the central aisle between seats 3 and 4. The seats are vacated in this order: 3 4 2 5 1 6.

Row							
1	1	2	3	Center	4	5	6
2	1	2	3	Aisle	**4**	5	6
3	1	2	3		4	5	6
.	1	2	3		4	5	6
.	1	2	3		4	5	6
.	1	2	3		4	5	6

If Dylan is sitting in row 2 seat 4, he wants to know how many people will get off the bus before him. In this situation, the entire first row will exit, followed by the person in seat 3 of row 2, and then Dylan exits after 7 people have gone before him.

As the possible numbers are enormous, they'll need a computer program to do this calculation!

Input: The first line is T (0 < T <= 100), the number of test cases. Each test case has 3 space separated integers, **K** (as defined above), **R** (the number of the row in which Dylan sits), and C (Dylan's seat number in that row).

1 <= K, R <= 10^8, 1 <= C <= 2 * K

Output: For each test case, print a single integer, the number of people who will disembark before Dylan. This value is guaranteed to fit in a 64-bit integer.

Sample input:

```
5
3 2 4
2 1 3
4 4 2
25 10 43
100000000 1 200000000
```

Sample output:

```
7
1
28
485
199999999
```